



**Certified Tech
Developer**

The Ultimate Degree

Objetivos

En el siguiente ejercicio vamos a crear una máquina virtual utilizando Vagrant.

¿Qué recibimos?

- Un vagrantfile
- El box de vagrant que vamos a utilizar, en caso de no haberlo realizado, en una carpeta de nuestro equipo ejecutamos la instrucción

```
vagrant box add debian/buster64
```

Instrucciones

Ejercicio 1 - De forma individual ejecutamos los siguiente pasos:

- 1) En una carpeta de nuestro equipo vamos a proceder a crear un archivo, el cual denominaremos vagrantfile (sin extensión)
- 2) Dentro de ese Vagrantfile pegaremos este texto (recomendamos usar un editor de texto avanzado, por ejemplo Notepad++)

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
# All Vagrant configuration is done below. The "2" in
# Vagrant.configure
# configures the configuration version (we support older
# styles for
# backwards compatibility). Please don't change it unless
# you know what
# you're doing.
Vagrant.configure("2") do |config|
```



```
config.vm.define "server" do |server|
  config.vm.box = "debian/buster64"
  server.vm.hostname = "server"
  server.vm.network "public_network"
end
end
```

Allí observamos en la línea 8 que definimos “server” como nombre de nuestra VM.

- 3) Una vez que tenemos nuestro Vagrantfile guardado, procedemos a iniciar la instanciación de la misma, con el comando

```
vagrant up
```

Es probable que un momento nos liste todos nuestros adaptadores de red , seleccionamos el correspondiente al de nuestra conectividad principal. Pasados unos minutos, tendremos nuestra VM instanciada con Vagrant disponible, para verificarlo ejecutamos

```
vagrant status
```

y deberíamos obtener lo siguiente:

```
C:\Users\edorio.AR-CARRERA-09\Documents\debian>vagrant status
Current machine states:

server                               running (virtualbox)
```

- 4) Ahora nos vamos a conectar a la instancia, para ello , en la misma carpeta donde instanciamos la VM, ejecutamos la sentencia

```
vagrant ssh server
```

Ello nos conectara a la instancia, por defecto y al no haber configurado usuarios, nos conectaremos con el usuario “vagrant”, pero podremos cambiar a root usando “su”; la contraseña por defecto para dicho usuario es “vagrant”



```
C:\Users\edorio.AR-CARRERA-09\Documents\debian>vagrant ssh server
Linux server 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 25 23:19:19 2021 from 10.0.2.2
vagrant@server:~$ su root
Password:
root@server:/home/vagrant#
```

SECUENCIA DE CONEXIÓN A LA VM

- 5) Una vez allí dentro, debemos obtener la ip de la VM, en este caso vamos a tener 3 adaptadores de red, nos interesa la número 3.
 - a) Una vez dentro de la VM, instalamos el paquete apache2, para ello usamos la instrucción “apt-get install apache2” como usuario root.
 - b) Finalmente, realizamos la misma prueba que la clase anterior, accediendo desde un navegador web a la página por defecto del Apache2.



Ejercicio 2 - De forma individual ejecutamos los siguiente pasos:

Ya vimos como Vagrant nos redujo sustancialmente la instanciación de una VM, pero aún hubo pasos como la instalación de Apache2 que lo hicimos de manera personalizada, es por ello que en esta parte de la ejercitación nos abocaremos a modificar algunos aspectos del Vagrantfile que harán más autónoma la instalación de paquetes y el cambio de algunas configuraciones.



- 1) En primer lugar, y como propósito de la prueba, vamos a destruir nuestra VM, con la siguiente sentencia:

```
vagrant destroy -f
```

De esta manera se forzará el cierre de la VM y la destrucción de la misma.

- 2) Modificaremos nuestro Vagrantfile, en primer lugar agregaremos las instrucciones para instalar el apache2, insertamos lo siguiente a partir de la línea 12 (antes del penúltimo end)

```
server.vm.provision "shell", inline: <<-SHELL
  apt-get update
  apt-get install -y apache2
SHELL
```

Luego validamos nuestro Vagrantfile y si está todo OK, la iniciamos.

```
vagrant validate
vagrant up
```

Nos conectamos a nuestra VM y allí nuevamente consultamos por la IP con "ip address", accedemos en nuestro navegador a dicha IP y veremos que ya se nos muestra por la pantalla de Apache2 por defecto, con lo cual las instrucciones que le agregamos al Vagrantfile se ejecutaron correctamente.

Con toda la mesa de trabajo debatan sobre las siguientes preguntas y contestarlas en conjunto:

- Tanto para el ejercicio 1 como para el ejercicio 2, describan con tus palabras lo que acaban de hacer.
- Describan con sus palabras cuál es la utilidad que se imaginan implementando Vagrant como herramienta.
- Determinen, de manera aproximada, cuánto tiempo demoraron en instanciar una VM con VirtualBox y luego esa configuración similar con Vagrant.
- Modifiquen una vez más el Vagrantfile, creen un HTML en sus máquinas (de manera sencilla, sino pueden tomar el código completo de este [HTML](#)) , nombrarlo como index.html, guardarlo en la misma carpeta que el Vagrantfile y luego agregamos las siguientes líneas



```
server.vm.provision "file", source: "index.html",  
destination: "index.html"  
server.vm.provision "shell", inline: "mv index.html  
/var/www/html/index.html"
```

Ahora averiguar la IP de la VM, navegan y ¿qué ven como página de inicio?